

# **EXHIBIT 44**

## **(Part 2 of 2)**

*Petition for Inter Partes Review of Patent No. 7,047,526*

translation associated with the phrase structure rule. (Clark Decl. ¶ 57; Ex. 1002 at 8:60-65; *see also id.* at 14:19-21, 17:47-58.)

The instruction to return an error that results from inputting an invalid word or incomplete sequence (e.g., inputting “delete” without a target pathname), or the translation function that results from inputting a complete and valid sequence of tokens (e.g., “delete /usr/extract/testing”), is what the ’526 patent refers to as a “command action value.” Such a command action value exists for each phrase structure rule in Martinez-Guerra, and for each generic command component of a given phrase structure rule, because as input words are processed, one may be invalid, resulting in an error, or all may be valid, leading ultimately to the satisfaction of a phrase structure rule, thus resulting in a translation according to the corresponding translation rule. (Clark Decl. ¶ 57.) Each legal choice in the parse state is thus an “element,” as described in the ’526 patent, having both a generic command component and a corresponding command action value. (*Id.* at 58.)

5. [1C.3] “the validating step including identifying one of the elements as a best match relative to the generic command; and”

In the generic command example discussed above, Martinez-Guerra discloses that the generic command “delete /usr/extract/testing” is validated by the parser-translator because it matches the legal ordering of tokens described by the

*Petition for Inter Partes Review of Patent No. 7,047,526*

phrase structure rule RULE1. (Clark Decl. ¶ 59; Ex. 1002 at 15:50-62.) In doing so, the parser-translator identifies the final element (“pathname-expert”) of the rule specification RULE1 as the “best match” relative to the generic command because it is the final input word necessary to satisfy the phrase structure rule, thus causing the parser-translator to issue a prescribed command. (Clark Decl. ¶ 59.) As Martinez-Guerra states, “[o]nce the current parse state includes a complete phrase as defined by the phrase structure rules encoded in grammar 37, translator 33 provides a translation.” (Ex. 1002 at 11:57-60.)

**6. [1D] “issuing a prescribed command of a selected one of the management programs according to the corresponding command format, based on the identified one element.”**

Martinez-Guerra discloses that when a phrase structure rule is satisfied by the sequence of words received in the input stream, the translation function associated with that phrase structure rule causes the parser-translator to perform the translation. (Clark Decl. ¶ 60-61; Ex. 1002 at 11:55-62, 14:10-21, 17:47-55, 20:61-64.) The translator outputs the translation in the format used by the appropriate software tool. (*Id.* at 13:30-33 (“[T]he translation functionality of parser-translator component 30 provides the capability of translating the now properly formed user language statements *to the form and syntax desired by a specific software tool.*”)) (emphasis added); *see also id.* at 15:59-62, 21:33-36.)

*Petition for Inter Partes Review of Patent No. 7,047,526*

For example, in the case of the generic command “delete /usr/extract/testing,” after matching the token corresponding to the final generic command word with the next legal choice in the parse state (“pathname-expert”), the parser-translator carries out the translation using the translation function associated with phrase structure rule “RULE1,” resulting in the prescribed UNIX command “rm /usr/extract/testing.” (*Id.* at 15:59-62.) One of ordinary skill in the art would recognize that “rm” is the UNIX shell command used to delete files in a computer system. (Clark Decl. ¶ 61.) Martinez-Guerra thus discloses issuing a prescribed command for removing a file in a selected one of the management programs (a UNIX shell) according to the corresponding command format for that program (“rm /usr/extract/testing”). (Ex. 1002 at 15:50-62, 16:5.)

7. [2A] **“The method of claim 1, wherein the generic command includes at least one input command word, the validating step including:”**

Martinez-Guerra discloses a generic command (*e.g.*, “delete /usr/extract/testing,”) which includes at least one input command word. (*Id.* at 15:50-62 (explaining that “delete” in RULE1 is a token “matching the string delete in the input stream such as input stream 36”).)

*Petition for Inter Partes Review of Patent No. 7,047,526*

8. [2B] “comparing each input command word to a command word translation table, configured for storing for each prescribed command word a corresponding token, for identification of a matching token; and”

The token recognizer subcomponent of the parser-translator in Martinez-Guerra extracts input words from the input stream “in accordance with dictionary entries and rules encoded in grammar 17 and in accordance with the current parse state.” (*Id.* at 9:35-38, *see also id.* at 11:51-55.) Martinez-Guerra explains that dictionary entries “encode information associated with tokens” (*id.* at 10:24-25), and “describe all the tokens that a parser-translator (*e.g.*, token recognizer 33 of parser-translator component 30) should recognize . . . .” (*Id.* at 14:16-18.) The ’526 patent describes the “command word translation table” in similar terms: it “includes all the command words 26 that are valid according to the generic syntax.” (Ex. 1001 at 3:44-45.) One of ordinary skill in the art would understand from these disclosures, and in view of Martinez-Guerra’s example of the “delete /usr/extract/testing” generic command, that the dictionary entries comprise a command word translation table used by the token recognizer to match input words with valid tokens found in the phrase structure rules. (Clark Decl. ¶ 63; *see also* Ex. 1002 at 14:55-59 (“A rule specification includes a sequence of grammar categories that define a legal sequence (or sequences) of tokens in the language. Tokens (read from the input stream) ‘match’ categories.”).)

*Petition for Inter Partes Review of Patent No. 7,047,526*

For example, when the user submits the “delete /usr/extract/testing” command, the first input word “delete” is matched with a “regular token” entry in the dictionary having the same content (“delete”). (Clark Decl. ¶ 64.) As Martinez-Guerra explains, a “regular token is a fixed sequence of characters (for example, the string ‘delete’).” (Ex. 1002 at 15:2-4.) The next (and final) input word the user submits is the pathname for the folder or file to be deleted. (Clark Decl. ¶ 64; Ex. 1002 at 15:57-16:5.) Martinez-Guerra explains that “pathname-expert” is used in place of the actual pathname entered by the user (*e.g.*, “/usr/extract/testing”). (Clark Decl. ¶ 64; Ex. 1002 at 15:6-15, 15:54-16:5.) When the input word “/usr/extract/testing” is received from the user, the parser-translator uses the dictionary entries to match the original input word with “pathname-expert.” (Clark Decl. ¶ 64; Ex. 1002 at 15:4-15, 15:63-16:5.) The parser-translator then uses the matching token “pathname-expert” from the dictionary entries *instead of* the original input word “/usr/extract/testing” when searching for a match in the command parse tree. (Clark Decl. ¶ 64; Ex. 1002 at 15:54-16:5.)

**9. [2C] “determining a presence of the matching token within the command parse tree for each input command word.”**

For each input command word the token recognizer subcomponent matches to a corresponding token in the dictionary entries (Ex. 1002 at 11:51-55), it determines whether that token is among the next legal choices in the parse states maintained by the parser. (*Id.* at 18:36-42, 18:66-19:7.) As discussed above, the

*Petition for Inter Partes Review of Patent No. 7,047,526*

parser-translator uses “internal data structures,” including “parse state data structures” and “internal representations of a grammar” to specify the next legal choices allowed by the grammar, and the phrase structure rules encoded in the grammar collectively constitute a command parse tree. (Clark Decl. ¶ 65; Ex. 1002 at 14:10-15, 18:31-48.) If the parser-translator determines that a token matches an expected next legal choice in the parse state, then it adds the token to the parse state, and the parser computes an updated list of next legal choices (*i.e.*, tokens that may legally follow the matched token in the command parse tree). (Clark Decl. ¶ 65; Ex. 1002 at 19:5-35.) Stated in terms of the ’526 patent, the parser-translator determines the presence of a matching token (*e.g.*, “delete”) within the command parse tree. (Clark Decl. ¶ 65.)

In the example discussed in the previous limitation, the parser-translator first determines the presence of the token corresponding to the input word “delete” in the command parse tree. (Clark Decl. ¶ 66; Ex. 1002 at 15:50-62, 18:31-48.) A matching token is found within the command parse tree because “delete” is the first token in the phrase structure rule RULE1. (Clark Decl. ¶ 66; Ex. 1002 at 15:57; 20:40-45, Fig. 10.) Having matched the first token, the parser-translator then determines the presence of the second (and final) token, “pathname-expert,” which corresponds to a valid file or folder pathname supplied by the user (“/usr/extract/testing”), as explained above. (Clark Decl. ¶ 66; Ex. 1002 at 15:50-

*Petition for Inter Partes Review of Patent No. 7,047,526*

16:5.) The “pathname-expert” token is also found in the command parse tree because it corresponds to the final token of the phrase structure rule RULE1. (Clark Decl. ¶ 66; Ex. 1002 at 15:57.) The parser-translator of Martinez-Guerra thus determines the presence of a matching token in the command parse tree for each input command word.

**10. [3] “The method of claim 2, wherein the determining step includes recursively traversing the command parse tree based on an order of the input command words for identification of the matching token within the identified one element.”**

As discussed above, the Martinez-Guerra token recognizer determines, for each input command word, the presence of a token in the phrase structure rules of the grammar, which for any finite high-level command language together constitute a command parse tree as described in the '526 patent. (Clark Decl. ¶ 67; Ex. 1002 at 18:31-40.) Martinez-Guerra teaches that “[t]oken recognizer 31 performs input matching against each of the valid next states corresponding to a parse state representation consistent with the input stream read so far.” (Ex. 1002 at 20:31-34.) As successive tokens are matched to potential next legal choices (*i.e.*, “elements”) in the representation of the current parse state (*id.* at 20:35-40), the parser-translator updates the current parse state. (Clark Decl. ¶ 67; Ex. 1002 at 20:42-45, 20:49-54.) This process of matching input tokens to next legal choices in the phrase structure rules is repeated for each word in the input stream, in order,

*Petition for Inter Partes Review of Patent No. 7,047,526*

until a complete phrase, as defined by a phrase structure rule, has been parsed. (Clark Decl. ¶ 67; 20:42-48.) Thus, Martinez-Guerra discloses processing the input stream based on an order (*i.e.*, left-to-right) of the input command words. (Clark Decl. ¶ 67.)

One of ordinary skill in the art would understand that this process “recursively traverses” the command parse tree in the same way as the ’526 patent. (Clark Decl. ¶ 68.) Figure 10 of Martinez-Guerra “illustrates the application of an input token to each parse state” (Ex. 1002 at 20:40-41), and this process repeats for each input token in the order received. (Clark Decl. ¶ 68; Ex. 1002 at 10:53-55, 18:62-65 (“Additional tokens are parsed from the input stream (*e.g.*, input stream 36) until a complete statement is defined in accordance with the particular grammar employed.”).) The same process is used in Figure 3 of the ’526 patent, where tokens are matched to elements in the command parse tree. (Clark Decl. ¶ 68.) Because Martinez-Guerra uses the same process for matching tokens to next legal choices in the command parse tree as the ’526 patent, it “recursively traverses” the command parse tree as claimed.

Moreover, even if the Board were to construe this claim limitation in accordance with the mathematical meaning of “recursive”—*i.e.*, “a function that calls itself”—Martinez-Guerra still renders this limitation obvious. Skilled artisans understood at the time of the alleged invention of the ’526 patent that repeating

*Petition for Inter Partes Review of Patent No. 7,047,526*

processes and recursive processes were known alternatives for performing tasks such as those claimed, and that a recursive process could be substituted for a repeating one with predictable results. (Clark Decl. ¶ 69.) In particular, recursive processing (in the stricter sense) has long been seen as particularly well-suited to trees. (*Id.*) Accordingly, it would have been obvious to a skilled artisan to modify the teachings of Martinez-Guerra to use a recursive process (*i.e.*, one that calls itself) rather than a repeating one as both Martinez-Guerra and the '526 patent disclose.

**11. [4] “The method of claim 3, wherein the issuing step includes issuing the prescribed command based on a corresponding command key specified for the matching token within the identified one element.”**

Martinez-Guerra discloses issuing the prescribed command based on a “translation function identifier” specified for the element that completes a phrase structure rule, such as the “pathname-expert” element of “RULE1.” (Clark Decl. ¶ 70; Ex. 1002 at 15:50-62, 16:11-16.) When the input stream contains a legal ordering of words according to a phrase structure rule, the translation function corresponding to that phrase structure rule is used to determine the translation. (Ex. 1002 at 16:15-16, 17:47-52, 20:61-64.) The output of this process is the prescribed command (*e.g.*, “rm /usr/extract/testing”). (Clark Decl. ¶ 70; Ex. 1002 at 15:50-62.)

*Petition for Inter Partes Review of Patent No. 7,047,526*

The “translation function identifier” specifies where a translation function corresponding to a matched phrase structure rule can be located. (Clark Decl. ¶ 70; Ex. 1002 at 16:11-16, 20:61-64.) It is thus a “command key,” as claimed in the ’526 patent. (Clark Decl. ¶ 70.) Martinez-Guerra explains that the translation function identifier “may take the form of . . . an index specifying a function number,” and “allows parser-translator component 30 (particularly token recognizer 31 and translator 33 subcomponents thereof) to invoke the corresponding . . . translation function in accordance with the grammar 37.” (Ex. 1002 at 16:16-24.)

Thus, in the example discussed in claim 1 above, when the final token “pathname-expert” (which is used in place of the input word “/usr/extract/testing” in the generic command “delete /usr/extract/testing” received from the user) is matched with the “pathname-expert” element in the portion of the command parse tree containing RULE1, a translation function identifier specified for RULE1 tells the parser-translator where to locate the translation function needed for the translator to issue the prescribed command “rm /usr/extract/testing.” (Clark Decl. ¶ 71.) Accordingly, the prescribed command is based on a corresponding command key specified for the matching token “pathname-expert” within the identified one element.

*Petition for Inter Partes Review of Patent No. 7,047,526*

12. [5] “The method of claim 4, wherein the issuing step further includes accessing a prescribed translator configured for converting the generic command according to the corresponding command format into the prescribed command based on the corresponding command key.”

Martinez-Guerra discloses accessing a prescribed translator (*e.g.*, translator 33) that is configured for converting the generic command according to the corresponding command format (*e.g.*, “*delete /usr/extract/testing*”) into the prescribed command (*e.g.*, “*rm /usr/extract/testing*”) based on the corresponding command key (*e.g.*, the translation function identifier corresponding to phrase structure rule RULE1). (Ex. 1002 at 17:51-58 (“A translation function is associated with a phrase structure rule and is invoked by translator 33.”); 15:50-62 (translating the generic command “*delete /usr/extract/testing*” to the prescribed command “*rm /usr/extract/testing*” based on the translation function corresponding to phrase structure rule RULE1).) Martinez-Guerra explains that, when a phrase structure rule is satisfied by the input words received, the translation corresponding to that rule is performed by a translator component of the parser-translator. (Clark Decl. ¶ 72; Ex. 1002 at 13:29-33, 15:24-28, 17:51-58, 20:61-64.) In carrying out such a translation, the prescribed translator that converts the generic command to the prescribed command is necessarily “accessed.” (Clark Decl. ¶ 72.) As discussed above with reference to claim 4, the “translation function identifier” specifies where a translation function corresponding to a matched phrase structure

*Petition for Inter Partes Review of Patent No. 7,047,526*

rule can be located (“the prescribed translator,” in the words of the’526 patent). (Clark Decl. ¶ 72; Ex. 1002 at 16:11-16, 20:61-64.) The accessed translator thus performs the translation based on the translation function identifier (*i.e.*, “command key”). (*See* discussion of claim 4 above; Clark Decl ¶ 72; Ex. 1002 at 20:61-64.)

13. [6/8] **“The method of claim [5/1], wherein the validating step including validating at least a portion of the generic command by identifying the one element having the best match relative to the portion of the generic command, the issuing step including issuing the prescribed command based on the identified one element corresponding to the portion of the generic command.”**

Claims 6 and 8 recite the same limitation, but depend respectively from claims 5 and 1. Because the claim limitations are identical, the analysis below applies to each of them.

As discussed above with reference to claim 1, the parser-translator validates the generic command “delete /usr/extract/testing” because the token recognizer determines, for each of the input words “delete” and “/usr/extract/testing” (where “/usr/extract/testing” is matched with the token “pathname-expert” in a dictionary entry), that the token corresponding to each input word is a next legal choice according to the phrase structure rule RULE1. (Clark Decl. ¶ 73.) By matching the final token (“pathname-expert”) to the corresponding element in RULE1, Martinez-Guerra discloses validating at least a portion of the generic command by

*Petition for Inter Partes Review of Patent No. 7,047,526*

identifying the one element having the best match relative to the portion of the generic command. (*Id.* ¶ 73.) Martinez-Guerra satisfies the limitation because validating the entire generic command “delete /usr/extract/testing” necessarily validates “at least a portion of” the generic command. Further, Martinez-Guerra discloses that if an input word is invalid or a command is incomplete—for example, if only the word “delete” is received without any pathname to define the target of the operation—that action may result in returning an error. (See Ex. 1002 at 19:3-4.) In this circumstance, the parser-translator identifies a “best match” for less than the entire command. (Clark Decl. ¶ 73.)

Martinez-Guerra further issues the prescribed command “rm /usr/extract/testing” based on the identified one element of the command parse tree (“pathname-expert”) corresponding to the portion of the generic command (“/usr/extract/testing”). As discussed above with reference to claim 1, “pathname-expert” is the final element in phrase structure rule RULE1. (*Id.* ¶ 74.) Martinez-Guerra explains that once the final token in the sequence of legal tokens for a phrase structure rule is matched, the translation rule corresponding to the phrase structure rule is used by the translator to output a translation. (Ex. 1002 at 9:40-44, 15:24-28, 17:47-52, 20:61-64.) Accordingly, when the “pathname-expert” token (which is used in place of the input word “/usr/extract/testing” in the generic command) is matched with the final element “pathname-expert” of RULE1,

*Petition for Inter Partes Review of Patent No. 7,047,526*

RULE1 is fully parsed, and the translation function corresponding to RULE1 is used to issue the prescribed command “rm /usr/extract/testing.” (Clark Decl. ¶ 74; Ex. 1002 at 20:61-64, 21:1-4.) Martinez-Guerra explains that this is accomplished by the translator substituting the original value of “pathname-expert” (the input word “/usr/extract/testing”) in the translation function (“rm <pathname-expert>”). (Ex. 1002 at 15:65-16:5.) The issued prescribed command is thus “rm /usr/extract/testing.” (*Id.* at 16:5.)

**14. [7/9] “The method of claim [6/8], further comprising executing the prescribed command within the corresponding selected one management program.”**

Claims 7 and 9 recite the same limitation, but depend respectively from claims 6 and 8. Because claims 7 and 9 recite the same limitation, and each depends from a claim that is rendered obvious by Martinez-Guerra as shown above, the analysis below applies to each of them.

It was obvious to one of skill in the art at the time of the alleged invention that, after the prescribed command is issued to the selected management program, the management program executes the command. As discussed with reference to claim 1, Martinez-Guerra discloses, *e.g.*, translating the generic command “delete /usr/extract/testing” into the prescribed command “rm /usr/extract/testing,” which is an executable command for removing the file located at “/usr/extract/testing” using a UNIX shell program. (Clark Decl. ¶ 75; Ex. 1002 at 15:50-62.) Martinez-

*Petition for Inter Partes Review of Patent No. 7,047,526*

Guerra’s “delete /usr/extract/testing” example is intended to illustrate the translation capability of the parser-translator and, accordingly, does not expressly state that the UNIX shell program executes this command. However, persons of ordinary skill in the art at the time of the alleged invention of the ’526 patent would have found it obvious from this example, Martinez-Guerra’s teachings, and their own knowledge, skill, and experience in the art, that the UNIX shell program would thereafter execute the prescribed “rm” command. (Clark Decl. ¶ 75.)

Martinez-Guerra states that an objective of the invention is to allow users to “specify complex . . . transformation statements in a high-level user language, . . . and to translate [those statements]. . . into logically and syntactically correct directives *for performing the desired data transformations or operations.*” (Ex. 1002 at 3:30-37; *see also* 4:4-9, 4:47-51) (emphasis added). One of ordinary skill in the art would understand from this disclosure that Martinez-Guerra intended that the translated directives (“prescribed commands”) were to be executed by the software tools that receive those directives. Indeed, a translated directive must be executed if it is to “perform the desired data transformation or operation.” (Clark Decl. ¶ 76.) If the command were not executed, there would be no purpose in carrying out the translation to begin with. (*Id.*)

In the example of the “delete /usr/extract/testing” command, the parser-translator translates the generic command “delete /usr/extract/testing” into the

*Petition for Inter Partes Review of Patent No. 7,047,526*

“logically and syntactically correct directive” “rm /usr/extract/testing” for the UNIX shell program. (*Id.* at ¶ 75; Ex. 1002 at 15:50-62.) One of ordinary skill in the art would understand from Martinez-Guerra’s disclosure that the purpose of such a translation is for the UNIX shell program to execute—or “perform[] the desired data transformation[] or operation[]” (Ex. 1002 at 3:36-37) according to—the prescribed command. (Clark Decl. ¶ 75.) Accordingly, these claims are invalid for obviousness.

**B. Claim 10 and dependent claims 11-13 are invalid as obvious over Martinez-Guerra**

**1. [10A] “A system configured for executing a plurality of management programs according to respective command formats, the system comprising:”**

As discussed above with reference to claim element 1A, Martinez-Guerra discloses the claimed system, wherein a parser-translator uses a “high-level user language” to interface with a plurality of software applications. (Ex. 1002, Abstract and 3:48-53.) It does so by translating statements (which can include generic commands) made in a high-level language to “logically and syntactically correct directives for performing the desired data transformations or operations” using one or more software tools (*e.g.*, prescribed commands). (*Id.* at 3:29-37.) The method is carried out by software executed on a computer system. (*Id.* at 9:10-13; *see also* Clark Decl. ¶ 79.)

*Petition for Inter Partes Review of Patent No. 7,047,526*

As also discussed above with reference to claim element 1A, Martinez-Guerra discloses a broad range of “management programs” with which the described parser-translator may be used. (*See* Clark Decl. ¶ 80.) Petitioners incorporate by reference and respectfully refer the Board to the discussion of claim element 1A for this limitation.

**2. [10B.1] “a parser having a command parse tree configured for validating a generic command received from a user,”**

The parser-translator of Martinez-Guerra, which includes a parser subcomponent (*id.* at 8:18-23, 9:24-26, Figs. 1-3), is configured to validate generic commands received from the user using phrase structure rules encoded in the grammar. (Clark Decl. ¶ 81.) As discussed above with reference to claim element 1C.1, for any finite high-level command language, Martinez-Guerra’s phrase structure rules—which Martinez-Guerra teaches may be represented as an “internal data structure[],” including an “internal representation[] of a grammar”—constitute a command parse tree because they are collectively a hierarchical data representation of the valid components of statements in a high-level language, whose elements specify at least one corresponding generic command component and a corresponding action value. (Clark Decl. ¶¶ 53, 81-82; Ex. 1002 at 18:31-35.)

Martinez-Guerra discloses a user interface that receives a statement from a user in a high-level language via an input stream. (Ex. 1002 at Fig. 1; 4:4-7; 9:15-

*Petition for Inter Partes Review of Patent No. 7,047,526*

23.) The parser-translator translates statements in that language into “statements or directives appropriate to a particular data processing application.” (*Id.* at 4:4-9.) One of ordinary skill in the art would understand that the high-level language described in Martinez-Guerra includes “generic commands” within the meaning of the ’526 patent, *i.e.*, a command that is an abstraction of a desired operation that is later translated into a directive for a specific software tool. (Clark Decl. ¶ 81; *see also* Ex. 1002, Abstract, 10:7-11, 13:30-33.) As discussed above, for example, Martinez-Guerra discloses that the parser-translator may receive a statement with the generic command, “delete /usr/extract/testing.” (Ex. 1002 at 15:50-62 (“the string delete in an input stream such as input stream 36”).)

As discussed with reference to claim element 1C.1 above, Martinez-Guerra teaches that the parser builds an “internal data structure[],” including an “internal representation[] of a grammar,” that identifies the “valid next states” according to phrase structure rules encoded in the grammar. (Clark Decl. ¶¶ 53, 81-82; Ex. 1002 at 18:31-48.) For a finite command language, these phrase structure rules encoded in the grammar collectively constitute a command parse tree within the meaning of the ’526 patent. (Clark Decl. ¶¶ 21-24, 53, 81-82.)

As input words are received from the user by the parser-translator, they are validated according to the phrase structure rules. As Martinez-Guerra explains, the token recognizer subcomponent of the parser-translator evaluates incoming words

*Petition for Inter Partes Review of Patent No. 7,047,526*

received from the user against these next legal choices in the phrase structure rules encoded in the grammar. (Clark Decl. ¶ 83; Ex. 1002 at 18:66-19:15; 19:25-35.) If the input word matches a next legal choice in the phrase structure rules, then it is validated because it is a permissible choice. (Clark Decl. ¶ 83; Ex. 1002 at 18:66-19:11.)

As Martinez-Guerra illustrates, if the input word “delete” is received by the parser-translator, it will match the initial (root level) legal choice for the specification of RULE1 (“delete” in the rule specification “delete pathname-expert”) because “delete” is a valid initial command term. (Clark Decl. ¶ 84; Ex. 1002 at 15:50-62; *see also* 19:25-29 (“[P]arser 32 begins with a first parse state and, consistent with phrase structure rules encoded in grammar 37, collects those tokens that are potentially valid next states corresponding to the first parse state.”).) Once validated, the token is accepted by the parser, and the parse state is updated to provide the token recognizer with a new list of next legal choices (*e.g.*, tokens that may permissibly follow the token “delete” in the internal data structure representing RULE1). (Clark Decl. ¶ 84; Ex. 1002 at 19:5-7, 19:16-25, 19:31-35.) If the user provides a valid pathname as the second input word, then the parser-translator validates the corresponding token as a next legal choice allowed by the phrase structure rules encoded in the grammar (*i.e.*, “pathname-expert” in the rule

*Petition for Inter Partes Review of Patent No. 7,047,526*

specification “delete pathname-expert”). (Clark Decl. ¶ 84; Ex. 1002 at 15:59-16:5, 20:40-49.)

The parser-translator thus uses a command parse tree configured to validate a generic command received from the user.

3. **[10B.2] “the command parse tree configured for specifying valid generic commands relative to a prescribed generic command format and having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value,”**

As explained above with reference to claim element 1C.1, Martinez-Guerra discloses a command parse tree configured for specifying valid generic commands (e.g., “delete /usr/extract/testing”) relative to a prescribed command format (e.g., as defined in RULE1, “delete pathname-expert.” (Ex. 1002 at 15:50-62; Clark Decl. ¶¶ 55, 83-84, 86.) Each token of a phrase structure rule, such as RULE1, is what the ’526 patent calls a “generic command component.” (Clark Decl. ¶ 86.) As further discussed above with reference to claim element 1C.2, each of these components of the phrase structure rules corresponds to an appropriate action that will occur if that component is a “best match” for the generic command. (Clark Decl. ¶¶ 87, 89.) Such a command action value exists for each generic command component of a given phrase structure rule because, as input words are parsed, they may be invalid, resulting in an error, or valid, leading ultimately to the satisfaction

*Petition for Inter Partes Review of Patent No. 7,047,526*

of a phrase structure rule, thus resulting in a translation according to the corresponding translation rule. (Clark Decl. ¶ 87.)

Each next legal choice in the parse state is thus an “element,” as described in the ’526 patent, having both a generic command component and a corresponding command action value. (Clark Decl. ¶ 88.)

**4. [10B.3] “the parser identifying one of the elements as a best match relative to the generic command; and”**

As explained with reference to claim element 1C.3 above, the token recognizer subcomponent of the parser-translator matches successive input words received from the user with the potential next legal choices in the parse state. (Clark Decl. ¶¶ 59, 89; Ex. 1002 at 18:38-42, 18:66-19:3.) When the input word matches a next legal choice that completes a phrase (*e.g.*, a valid pathname matching the next legal choice “pathname-expert” in the rule specification RULE1 (Ex. 1002 at 15:50-62)), the parser identifies that element as a best match for the generic command. (Clark Decl. ¶ 89; Ex. 1002 at 20:45-48.)

**5. [10C.1] “a plurality of translators configured for issuing commands for the management programs according to respective command formats,”**

The parser-translator of Martinez-Guerra is configured for issuing commands for a management program according to respective command formats. Martinez-Guerra discloses that multiple implementations of the parser-translator are possible, including one in which “multiple instances of a parser-translator

*Petition for Inter Partes Review of Patent No. 7,047,526*

component, each with a corresponding grammar, provide multiple software applications with a common interface to high-level user language statements.” (Ex. 1002 at 3:43-46.) An implementation of multiple parser-translator components includes a “plurality of translators.” (Clark Decl. ¶ 90.) Likewise, Martinez-Guerra describes an embodiment “wherein operation of [a] single parser-translator component is suitably defined for each software application using a corresponding grammar encoding” (Ex. 1002 at 3:51-53), which a POSA would understand is functionally equivalent to a plurality of translators. (Clark Decl. ¶ 90.)

As Martinez-Guerra explains, the disclosed translators are configured for issuing commands for management programs according to respective command formats. This is why each of the instances of parser-translators has “a corresponding grammar.” (Ex. 1002 at 3:53; *see also id.* at 3:43-53; Clark Decl. ¶ 91.) For example, Martinez-Guerra discloses that a translator subcomponent of the parser-translator issues an appropriate translation of the generic command “delete /usr/extract/testing” for the UNIX environment as “rm /usr/extract/testing.” (Clark Decl. ¶ 91; Ex. 1002 at 15:59-16:5.) This is a respective command format of the UNIX environment because “rm” is a UNIX shell command to delete the target file. (Clark Decl. ¶ 91.)

*Petition for Inter Partes Review of Patent No. 7,047,526*

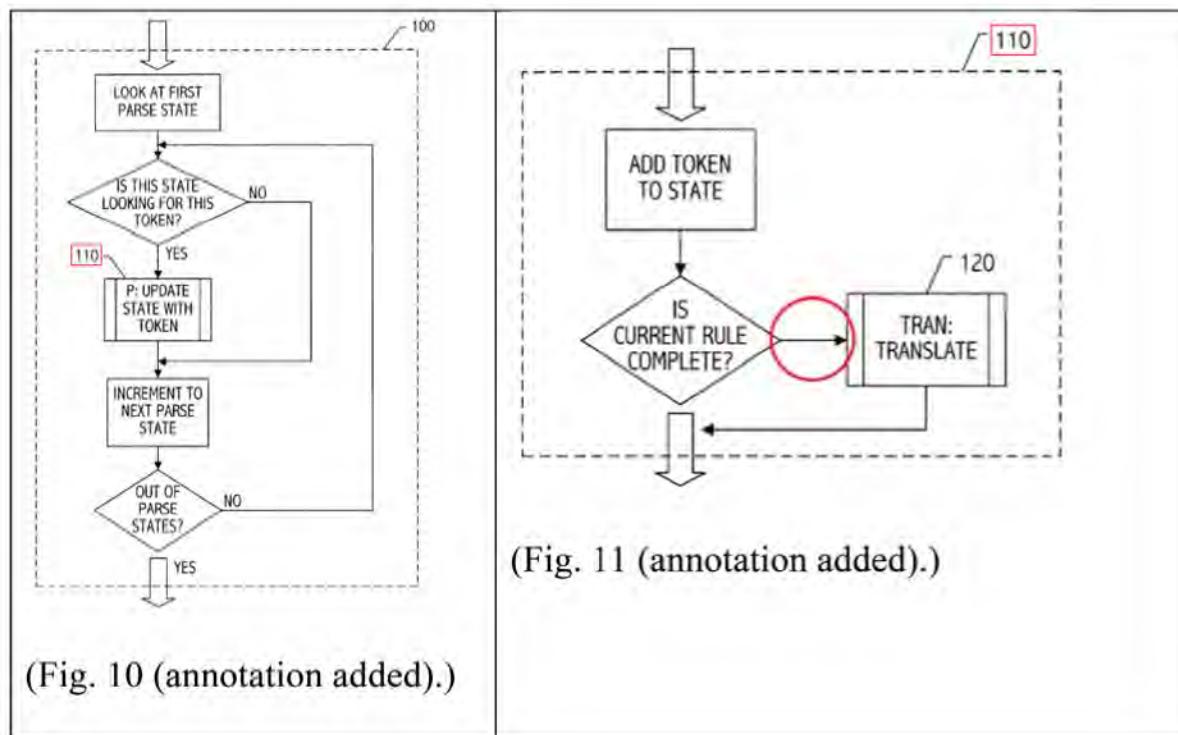
6. [10C.2] “the parser outputting a prescribed command to a selected one of the translators based on the identified one element.”

Martinez-Guerra discloses that, once a phrase structure rule is complete, which occurs when the final element of that rule is matched with a token from the input stream (Ex. 1002 at 20:45-48), the parser outputs a prescribed command to a translator based on that element. (Clark Decl. ¶ 92.) Figures 10 and 11 of Martinez-Guerra illustrate this process. Figure 10 discloses the process for determining whether an input token matches the next legal choice for one of the parse states maintained by the parser. (Clark Decl. ¶¶ 92; 20:40-42.) If the input token is found in the portion of the command parse tree containing that phrase structure rule, then the parser updates the parse state by adding the token using process 110 of Figure 10. (Clark Decl. ¶ 92; Ex. 1002 at 20:49-58.)

Process 110 is illustrated in Figure 11, wherein the parser adds the token to the parse state and determines whether that addition has completed a phrase structure rule (*e.g.*, in the example discussed above, the final token, “pathname-expert,” has been received). (Clark Decl. ¶ 92; Ex. 1002 at 20:42-45.) If the token completes the rule, Figure 11 illustrates that the parser outputs a prescribed command, identified by the arrow directed to box 120, to a translator (“TRAN:”). (Clark Decl. ¶ 92; Ex. 1002 at 20:40-49.) Box 120, illustrated in Figure 12, is the

*Petition for Inter Partes Review of Patent No. 7,047,526*

process undertaken by a translator subcomponent of the parser-translator. (Clark Decl. ¶ 92; Ex. 1002 at 20:45-48, Fig. 12.)



Because the command represented in Figure 11 is output by the parser once the final element of a phrase structure rule has been matched, it is output “based on the identified one element.” (Clark Decl. ¶ 93.)

7. [11A.1] “The system of claim 10, wherein the parser further comprises a command word translation table configured for storing for each prescribed command word a corresponding token for identification of a matching token,”

Martinez-Guerra discloses that the “parser-translator 10 receives an input stream 6 from which token recognizer 11 extracts tokens in accordance with dictionary entries and rules encoded in the grammar 17 and in accordance with the

*Petition for Inter Partes Review of Patent No. 7,047,526*

current parse state.” (Ex. 1002 at 9:35-40.) As illustrated in Figures 1-3, the grammar—which Martinez-Guerra explains includes the dictionary entries—is depicted as a separate functional unit from the parser-translator. (*Id.* at Figs. 1-3, 14:8-18; Clark Decl. ¶ 94.) The “dictionary entries” used by the parser-translator “describe all the tokens that a parser-translator . . . should recognize,” (Ex. 1002 at 14:16-18) and as explained above in relation to claim element 2B, parallel the ’526 patent’s description of the command word translation table (“includ[ing] all the command words 26 that are valid according to the generic syntax” (Ex. 1001 at 3:44-46)).

One of ordinary skill in the art would understand from these disclosures, and in view of Martinez-Guerra’s example of the “delete /usr/extract/testing” generic command, that the dictionary entries are a command word translation table coextensive with the operation of the parser-translator. (Clark Decl ¶ 94.) Martinez-Guerra expressly notes that one of ordinary skill in the art would understand that the boundaries between parser-translator subcomponents and their respective functionalities are variable and subject to compositions other than those described (Ex. 1002 at 9:26-31, 22:37-42), and one of skill in the art would appreciate that the dictionary entries may be part of the parser-translator. (Clark Decl. ¶ 94.)

*Petition for Inter Partes Review of Patent No. 7,047,526*

As explained with reference to claim element 2B above, when the user submits the “delete /usr/extract/testing” command, the first input word “delete” is matched with a “regular token” entry in the dictionary having the same content (“delete”). (Clark Decl. ¶¶ 64, 95.) As Martinez-Guerra explains, a “regular token” is a fixed sequence of characters (for example, the string ‘delete’).” (Ex. 1002 at 15:2-4.) The next (and final) input word the user submits is the pathname for the folder or file to be deleted (“/usr/extract/testing” in the example). (Clark Decl. ¶ 64, 95; Ex. 1002 at 15:57-16:5.) Martinez-Guerra explains that the dictionary entries are used to match the original input word with “pathname-expert.” (Clark Decl. ¶¶ 64, 95; Ex. 1002 at 15:4-15, 15:63-16:5.) The parser-translator then uses the matching token “pathname-expert” instead of the original input word “/usr/extract/testing” when searching for a match in the command parse tree. (Clark Decl. ¶¶ 64, 95.)

**8. [11A.2] “the parser configured for determining a presence of the matching token within the command parse tree for each input command word.”**

Martinez-Guerra discloses that the parser-translator determines a presence of the matching token within the command parse tree for each input word. In the example discussed above, the parser-translator first determines a presence of the token corresponding to the input word “delete” in the command parse tree, which reflects the initial parse state. (Clark Decl. ¶¶ 66, 96; Ex. 1002 at 15:50-62, 18:31-

*Petition for Inter Partes Review of Patent No. 7,047,526*

48.) A matching token is found within the command parse tree because “delete” is the first token in the phrase structure rule RULE1. (Clark Decl. ¶¶ 66, 96; Ex. 1002 at 15:57; 20:40-45, Fig. 10.) Having matched the first token, the parser-translator then determines the presence of the second (and final) token, which Martinez-Guerra explains is “pathname-expert,” corresponding to a valid file pathname supplied by the user, as explained above. (Clark Decl. ¶¶ 66, 96; Ex. 1002 at 15:50-16:5, 20:40-45.) The presence of “pathname-expert” is also found in the command parse tree because it corresponds to the final token of the phrase structure rule RULE1. (Clark Decl. ¶¶ 66, 96; Ex. 1002 at 15:57.) The parser-translator of Martinez-Guerra thus determines the presence of a matching token in the command parse tree for each input command word.

**9. [12] “The system of claim 11, wherein the parser recursively traverses the command parse tree based on an order of the input command words for identification of the matching token within the identified one element.”**

Claim 12 recites the same method as claim 3, discussed above, but specifies that the method is performed by the parser of independent claim 10. As discussed with reference to claim 3, Martinez-Guerra teaches that the claimed method is performed by the disclosed parser-translator. (Clark Decl. ¶¶ 67-69, 97; Ex. 1002 at 20:31-34, 20:38-45.) Because the method performed by the parser of claim 12 is identical to that recited in claim 3, the analysis for this claim is the same as for

*Petition for Inter Partes Review of Patent No. 7,047,526*

claim 3. Accordingly, Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim 3, above.

**10. [13] “The system of claim 12, wherein the parser validates at least a portion of the generic command by identifying the one element having the best match relative to the portion of the generic command.”**

As discussed above with reference to claim 10, the parser-translator validates the generic command “delete /usr/extract/testing” in the example given at column 15:50-62 because the token recognizer portion of the parser-translator determines, for each of the input words “delete” and “/usr/extract/testing” (where “/usr/extract/testing” is matched with the token “pathname-expert” in a dictionary entry), that the token corresponding to each input word is a next legal choice according to phrase structure rule RULE1. (Clark Decl. ¶¶ 89, 98.) By matching the final token (“pathname-expert”) to the corresponding element in RULE1 (“pathname-expert”), Martinez-Guerra discloses that the parser-translator validates at least a portion of the generic command by identifying the one element having the best match relative to the portion of the generic command. (Clark Decl. ¶ 98; Ex. 1002 at 20:31-34.) Martinez-Guerra satisfies the limitation because validating the entire generic command “delete /usr/extract/testing” necessarily validates “at least a portion of” the generic command. (Clark Decl. ¶ 98.) Further, Martinez-Guerra discloses that if an input word is invalid or a command is incomplete—for example, if only the word “delete” is received without any pathname to define the

*Petition for Inter Partes Review of Patent No. 7,047,526*

target of the operation—that action may result in returning an error. (See Ex. 1002 at 19:3-4.) In this circumstance, the parser-translator identifies a “best match” for less than the entire command. (Clark Decl. ¶ 98.)

**C. Claim 14 and dependent claims 15-22 are invalid as obvious over Martinez-Guerra**

1. [14A] “A computer readable medium having stored thereon sequences of instructions for executing a plurality of management programs according to respective command formats, the sequences of instructions including instructions for performing the steps of:”

Claim 14 recites the same method steps as claim 1, discussed above, but is drafted in the form of a computer-readable medium for performing the claimed steps. Martinez-Guerra discloses that the invention may be carried out by instructions embodied in a computer-readable medium. (Ex. 1002 at 4:52-56, 5:51-53, 13:55-14:7, 22:45-57, 24:14-34 (claims 16-17), 25:61-26:10 (claims 32-33).) Because the limitations of claim 14 following the preamble are identical to those of claim 1, claim 14 is obvious for the same reasons articulated as to claim 1.

Accordingly, Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim elements 1A-D above.

**2. Dependent claims 15-19 and 21 are invalid as obvious over Martinez-Guerra**

Dependent claims 15-19 and 21 recite the same limitations as dependent claims 2-6 and 8, respectively, but are drafted as dependent from computer-readable medium claim 14. As discussed above, Martinez-Guerra discloses that the

*Petition for Inter Partes Review of Patent No. 7,047,526*

invention may be carried out by instructions embodied in a computer-readable medium. (*Id.* at 4:52-56, 5:51-53, 13:55-14:7, 22:45-57, 24:14-34 (claims 16-17), 25:61-26:10 (claims 32-33).) Because the limitations of dependent claims 15-19 and 21 are otherwise identical to dependent claims 2-6 and 8, claims 15-19 and 21 are obvious for the same reasons articulated as to claims 2-6 and 8, respectively. Accordingly, Petitioner incorporates by reference and respectfully refers the Board to the discussion of claims 2-6 and 8 above.

3. [20/22] “The medium of claim [19/21], further comprising instructions for performing the step of executing the prescribed command within the corresponding selected one management program.”

Claims 20 and 22 recite the same limitation as claims 6 and 8, discussed above, but depend respectively from claims 19 and 21. Both claims 20 and 22 ultimately depend from computer-readable medium claim 14. As discussed with reference to claim 14, Martinez-Guerra discloses that the invention may be carried out by instructions embodied in a computer-readable medium. (Ex. 1002 at 4:52-56, 5:51-53, 13:55-14:7, 22:45-57, 24:14-34 (claims 16-17), 25:61-26:10 (claims 32-33).) Because claims 20 and 22 are otherwise identical to claims 7 and 9, the analysis for those claims applies equally to claims 20 and 22. Accordingly, Petitioner incorporates by reference and respectfully refers the Board to the discussion of claims 7 and 9 above.

*Petition for Inter Partes Review of Patent No. 7,047,526***D. Claim 23 and dependent claims 24-26 are invalid as obvious over Martinez-Guerra**

- 1. [23A] “A system configured for executing a plurality of management programs according to respective command formats, the system comprising:”**

The preamble of claim 23 recites the same system as claim 10, discussed above. Accordingly, Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim element 10A above.

- 2. [23B.1] “means for validating a generic command received from a user,”**

Martinez-Guerra discloses a parser-translator (*e.g.*, parser-translator 10 of Figure 1 and parser-translator 30 of Figure 3) that validates a generic command received from a user. (Clark Decl. ¶ 109.) As explained above with reference to claim element 10B.1, the discussion of which is incorporated by reference, Martinez-Guerra discloses that the parser-translator validates input words—including words for a generic command such as “delete /usr/extract/testing”—received from a user via an input stream. (*See* Clark Decl. ¶¶ 81-85; Ex. 1002 at Figs. 1, 3; 4:4-9; 9:15-23, 15:54-16:5.) The parser-translator of Martinez-Guerra thus includes the claimed “means for validating.”

*Petition for Inter Partes Review of Patent No. 7,047,526*

3. [23B.2] “the validating means configured for specifying valid generic commands relative to a prescribed generic command format and having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value,”

Martinez-Guerra discloses that the parser-translator uses a grammar defining “a set of phrase structure rules and associated translation rules that define the syntactic order of a source and target language . . .” (Ex. 1002 at 7:57-59, *see also id.* at 16:15-16) and dictionary entries “that describes all the tokens that a parser-translator (*e.g.*, token recognizer 31 of parser-translator component 30) should recognize” (*id.* at 14:16-18.) As explained above with reference to claim element 10B.2, the discussion of which is incorporated by reference, the parser-translator is thus configured for specifying valid generic commands relative to a prescribed generic command format. (Clark Decl. ¶¶ 86-88, 110.)

As further discussed above with reference to claim element 10B.2, the “internal data structure[]” representation of the phrase structure rules created and maintained by the parser-translator component has elements specifying generic command components (*e.g.*, “delete,” “pathname-expert”) and a corresponding command action value (*e.g.*, returning an error or performing a translation according to the completed phrase structure rule’s corresponding translation function) for each token of a phrase structure rule. (*See* Clark Decl. ¶ 87, 111; Ex. 1002 at 8:60-65, 14:11-12, 14:19-21, 15:50-62, 16:15-16, 17:47-58, 18:31-35,

*Petition for Inter Partes Review of Patent No. 7,047,526*

19:3-4.) For these reasons and those specified above as to claim element 10B.2, Martinez-Guerra discloses this limitation.

**4. [23B.3] “the validating means identifying one of the elements as a best match relative to the generic command; and”**

As explained above with reference to claim element 10B.3, the discussion of which is incorporated by reference, Martinez-Guerra discloses that the parser-translator identifies the element corresponding to the final token needed to complete a phrase structure rule (*e.g.*, “pathname-expert”) as the best match relative to the generic command (*e.g.*, “delete /usr/extract/testing”). (Clark Decl. ¶¶ 89, 112.) Because “pathname-expert” is the last token needed to satisfy the phrase structure rule “RULE1,” it is the best match relative to the generic command “delete /usr/extract/testing.” (*Id.*) For these reasons and those specified above as to claim 10B.3, Martinez-Guerra discloses this limitation.

**5. [23C.1] “a plurality of translators configured for issuing commands for the management programs according to respective command formats,”**

This limitation is identical to that of claim element 10C.1, discussed above, and the analysis is accordingly the same. Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim element 10C.1 above.

*Petition for Inter Partes Review of Patent No. 7,047,526*

6. [23C.2] “**the validating means outputting a prescribed command to a selected one of the translators based on the identified one element.**”

This limitation is identical to that of claim element 10C.2, discussed above, except that claim element 23C.2 requires the claimed outputting to be performed by “the validating means” rather than “the parser,” as in claim 10. As discussed above, the “validating means” includes the parser-translator and, accordingly, the analysis for this limitation is the same as claim element 10C.2. Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim element 10C.2 above.

7. [24] “**The system of claim 23, wherein the validating means comprises a command word translation table configured for storing for each prescribed command word a corresponding token for identification of a matching token, the validating means configured for determining a presence of the matching token for each input command word.**”

Claim 24 recites the same system as claim 11, discussed above, but requires the limitation be performed by “validating means” rather than the “parser,” as in claim 11. As discussed in independent claim 23, the claimed “validating means” includes the same parser-translator discussed in claim 10 and, accordingly, the analysis for this claim is the same as for claim 11. Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim 11 above.

*Petition for Inter Partes Review of Patent No. 7,047,526*

8. [25] “**The system of claim 24, wherein the validating means recursively validates each input command word based on an order of the input command words for identification of the matching token within the identified one element.**”

Claim 25 recites the same system as claim 12, discussed above, but requires the limitation be performed by “validating means” rather than the “parser,” as in claim 12. As discussed in independent claim 23, the claimed “validating means” includes the same parser-translator discussed in claim 10 and, accordingly, the analysis for this claim is the same as for claim 12. Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim 12 above.

9. [26] “**The system of claim 25, wherein the validating means validates at least a portion of the generic command by identifying the one element having the best match relative to the portion of the generic command.**”

Claim 26 recites the same system as claim 13, discussed above, but requires the limitation be performed by the “validating means” rather than the “parser,” as in claim 13. As discussed with respect to independent claim 23, however, the claimed “validating means” includes the same parser-translator discussed in claim 10 and, accordingly, the analysis for this claim is the same as for claim 13. Petitioner incorporates by reference and respectfully refers the Board to the discussion of claim 13 above.

## **IX. CONCLUSION**

Petitioner respectfully requests institution of an *inter partes* review and cancellation of claims 1-26 of the ’526 patent.

*Petition for Inter Partes Review of Patent No. 7,047,526*

Respectfully submitted,

DATED: November 4, 2015

/s/ Eugene M. Paige

Eugene M. Paige, Registration No. 55,519  
Counsel for Petitioner, Arista Networks,  
Inc.

Keker & Van Nest LLP  
633 Battery Street  
San Francisco, CA 94111  
(415) 391-5400 (telephone); (415) 397-  
7188 (fax)

*Petition for Inter Partes Review of Patent No. 7,047,526*

**CERTIFICATION OF SERVICE (37 C.F.R. §§ 42.6(e), 42.105(a))**

The undersigned hereby certifies that a copy of the foregoing **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 7,047,526**, the accompanying Power of Attorney, list of exhibits for Petition for *Inter Partes* Review, and all associated exhibits were served in their entireties on the following parties via FedEx®:

Cisco c/o Leon R. Turkevich  
Manelli Selter PLLC  
10560 Main Street, Suite PH6  
Fairfax, VA 22030

*Address of record as listed on PAIR*

Sean Sang-Chul Pak  
John M. Neukom  
Matthew Cannon  
Quinn Emanuel Urquhart & Sullivan LLP  
50 California Street, Floor 22  
San Francisco, CA 94111  
seanpak@quinnmanuel.com  
johnneukom@quinnmanuel.com  
Cisco-Arista@quinnmanuel.com

*Additional address known to Petitioner as likely to effect service*

Adam R. Alper  
Kirkland & Ellis LLP  
555 California Street  
San Francisco, CA 94104  
aalper@kirkland.com

*Additional address known to Petitioner as likely to effect service*

/s/ Eugene M. Paige

Eugene M. Paige, Registration No. 55,519  
Counsel for Petitioner, Arista Networks, Inc.

Keker & Van Nest LLP  
633 Battery Street  
San Francisco, CA 94111  
(415) 391-5400 (tel); (415) 397-7188 (fax)